



# Programación de Microservicios con Spring Boot y Red Hat SSO

Nivel Avanzado

Instructor: Carlos Carreño  
ccarrenovi@gmail.com



# Modulo 9 Protección de microservicios en cadena

**Objetivo:** Introducción a Red Hat SSO y su integración con el desarrollo de aplicaciones con Spring Boot

Duración: 1h



# Propagación de tokens entre servicios

Cuando un microservicio A recibe una petición autenticada (con token JWT) y necesita llamar a un microservicio B, debe **reenviar el token JWT original** para mantener el contexto de seguridad (identidad, claims, roles).

Flujo:

- ✓ Obtiene el JWT original desde el contexto de seguridad (**SecurityContextHolder**).
- ✓ Lo agrega como header "Authorization: Bearer <token>" a la petición que el microservicio A hace al microservicio B.
- ✓ Así el microservicio B puede validar el mismo token sin perder identidad.



# Transmisión segura de encabezados de autenticación

```
import feign.RequestInterceptor;
import feign.RequestTemplate;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.oauth2.server.resource.authentication.JwtAuthenticationToken;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class FeignClientConfig {

    @Bean
    public RequestInterceptor requestInterceptor() {
        return template -> {
            var authentication = SecurityContextHolder.getContext().getAuthentication();
            if (authentication instanceof JwtAuthenticationToken jwtAuth) {
                String tokenValue = jwtAuth.getToken().getTokenValue();
                template.header("Authorization", "Bearer " + tokenValue);
            }
        };
    }
}
```



# Invocación desde el controlador

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController

public class OrderController {

    private final UserServiceClient userServiceClient;

    public OrderController(UserServiceClient userServiceClient) {
        this.userServiceClient = userServiceClient;
    }

    @GetMapping("/api/order/user-info")
    public String getUserInfoFromUserService() {
        return userServiceClient.getUserInfo();
    }
}
```



# Lab

- Ejercicio: Microservicio consumidor que reenvía el token

